Q1 a) What is DNS? Explain its working with the example. [5]
Ans:
DNS: Domain Name Servers (DNS) are the Internet’s equivalent of a phone book. They maintain a directory of domain names and translate them to Internet Protocol (IP) addresses. This is necessary because, although domain names are easy for people to remember, computers or machines, access websites based on IP addresses.
Example:
The client (resolver) can ask for a recursive answer from a name server. This means that the resolver expects the server to supply the final answer. If the server is the authority for the domain name, it checks its database and responds. If the server is not the authority, it sends the request to another server (the parent usually) and waits for the response. If the parent is the authority, it responds; otherwise, it sends the query to yet another server. When the query is finally resolved, the response travels back until it finally reaches the requesting client. This is called recursive resolution and is shown in Figure below.

Q1 b) I was downloading an image image1.jpg using the following URL on 2\textsuperscript{nd} November, 2015. http://www.stockphoto.com/images/image1.gif. Show HTTP request and response messages for getting the image first time.[5]
Ans:
HTTP Request Message
GET http://www.stockphoto.com/images/image1.gif HTTP/1.1
Host: www.someschool.edu
User-agent: Mozilla/4.0
Connection: close
HTTP Response Message
HTTP/1.1 200 OK
Connection close
Date: 2 November 2015 04:36:25 GMT
Server: Apache/1.3.0 (Unix)
Connection: close
Expires: 2 November, 2015 05:36:25 GMT
Cache-Control: max-age=3600, public

Q2 a) What is difference between persistent and non-persistent HTTP? Also explain HTTP message format. [4]

Ans:

<table>
<thead>
<tr>
<th>Non-Persistent HTTP</th>
<th>Persistent HTTP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default for HTTP/1.0</td>
<td>Default for HTTP/1.1</td>
</tr>
<tr>
<td>server parses request, responds, and closes TCP connection</td>
<td>on same TCP connection: server, parses request, responds, parses new request,..</td>
</tr>
<tr>
<td>2 RTTs to fetch each object</td>
<td>Client sends requests for all referenced objects as soon as it receives base HTML.</td>
</tr>
<tr>
<td>Each object transfer suffers from slow start</td>
<td>Fewer RTTs and less slow start.</td>
</tr>
</tbody>
</table>

HTTP messages include requests from client to server and responses from server to client which will have the following format:

**Format of an HTTP Request**

It has three main components, which are:

- **HTTP Request Method, URI, and Protocol Version:** it contains the HTTP Request method being used for that particular request, the URI, and the HTTP protocol name with the version being used. It may look like 'GET /servlet/jspName.jsp HTTP/1.1' where the request method being used is 'GET', the URI is '/servlet/jspName.jsp', and the protocol (with version) is 'HTTP/1.1'.

- **HTTP Request Headers** - this section of an HTTP Request contains the request headers, which are few of these headers are: Content-Type, User-Agent, Accept-Encoding, Content-Length, Accept-Language, Host.

- **HTTP Request Body** - this part contains the actual request being sent to the HTTP Server.

**Format of an HTTP Response**

Similar to an HTTP Request, an HTTP Response also has three main components are:

- **Protocol/Version, Status Code, and its Description** - the very first line of a valid HTTP Response consists of the protocol name, it's version, status code of the request, and a short description of the status code. A status code of 200 means 'OK'. '404' means the file requested was not found at the HTTP Server.

- **HTTP Response Headers** - HTTP Responses are formed at the server machine. Few of these HTTP Response headers are: Server, Content-Type, Last-Modified, Content-Length, etc.
Q2b) what is DHCP? What are different types of DHCP messages” Explain.[5]

Ans:

**DHCP is Dynamic Host Configuration Protocol used for**

- Automatic IP addresses are assigned on demand
- Avoid manual IP configuration
- Support mobility of laptops
- Support temporary allocation of IP addresses

**DHCP Messages:**

<table>
<thead>
<tr>
<th>Value</th>
<th>Message Type</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DHCPDISCOVER</td>
<td>DHCP client sends a DHCP Discover broadcast on the network for finding a DHCP server</td>
</tr>
<tr>
<td>2</td>
<td>DHCPOFFER</td>
<td>DHCP servers on a network that receive a DHCP Discover message respond with a DHCP Offer message, which offers the client an IPv4 address lease.</td>
</tr>
<tr>
<td>3</td>
<td>DHCPREQUEST</td>
<td>Clients accept the first offer received by broadcasting a DHCP Request message for the offered IPv4 address.</td>
</tr>
<tr>
<td>4</td>
<td>DHCPDECLINE</td>
<td>If the DHCP client determines the offered TCP/IP configuration parameters are invalid, it sends a DHCPDecline packet to the server.</td>
</tr>
<tr>
<td>5</td>
<td>DHCPACK</td>
<td>The server accepts the request by sending the client a DHCP Acknowledgment message.</td>
</tr>
<tr>
<td>6</td>
<td>DHCPNAK</td>
<td>If the IPv4 address requested by the DHCP client cannot be used, the DHCP server responds with a DHCPNak.</td>
</tr>
<tr>
<td>7</td>
<td>DHCPRELEASE</td>
<td>A DHCP client sends a DHCPRelease packet to the server to release the IPv4 address.</td>
</tr>
<tr>
<td>8</td>
<td>DHCPINFORM</td>
<td>DHCPInform is used by DHCP clients to obtain DHCP options.</td>
</tr>
</tbody>
</table>

Q3a) What causes Silly window syndrome? How is it solved? Explain with an example. [5]

Ans:

**Silly Window syndrome:** It is a problem caused by poorly implemented TCP flow control. A serious problem can arise in the sliding window operation when the sending application program creates data slowly, the receiving application program consumes data slowly, or both. The name of this problem is due to the window size shrinking to a "silly" value.
Sender Side Solution: - Nagle's algorithm

- Purpose is to allow the sender TCP to make efficient use of the network, while still being responsive to the sender applications.
- **Idea:**
  - If application data comes in byte by byte, send first byte only. Then *buffer all application data till until ACK for first byte comes in.*
  - If network is slow and application is fast, the second segment will contain a lot of data.
  - Send second segment and buffer all data till ACK for second segment comes in.
  - An exception to this rule is to always send (not wait for ACK) if enough data for half the receiver window or MSS(Maximum segment size) is accumulated.

Receiver-Side-solution: - Clark's algorithm

- Purpose is to prevent the receiver from sending a window update for 1byte.
- **Idea:**
  - Receiver is forced to wait until it has a decent amount of space available
  - The receiver should not send a window update until it can handle the maximum segment size it declared when the connection was established or until its buffer is half empty, whichever is smaller

Q3 b) what is Socket? Which are various socket primitives used in client and server communication? [5]

**Ans:** Socket: A socket is one endpoint of a two-way communication link between two programs running on the network. A socket is bound to a port number so that the TCP layer can identify the application that data is destined to be sent to. An endpoint is a combination of an IP address and a port number.
Socket Primitives:

<table>
<thead>
<tr>
<th>Primitive</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOCKET</td>
<td>Create a new communication end point</td>
</tr>
<tr>
<td>BIND</td>
<td>Attach a local address to a socket</td>
</tr>
<tr>
<td>LISTEN</td>
<td>Announce willingness to accept connections; give queue size</td>
</tr>
<tr>
<td>ACCEPT</td>
<td>Block the caller until a connection attempt arrives</td>
</tr>
<tr>
<td>CONNECT</td>
<td>Actively attempt to establish a connection</td>
</tr>
<tr>
<td>SEND</td>
<td>Send some data over the connection</td>
</tr>
<tr>
<td>RECEIVE</td>
<td>Receive some data from the connection</td>
</tr>
<tr>
<td>CLOSE</td>
<td>Release the connection</td>
</tr>
</tbody>
</table>

Q4a) My friend Balu says that TCP header having a checksum field is redundant as IP header already has a checksum. Do you think Balu is right? Yes/No. Validate your claim with relevant examples. [6]
Ans:
Balu is not right; TCP header checksum field is not redundant with IP header checksum.
Reasons:
- IP checksum is only performed for the ipv4 header (first 20 bytes) not for the payload.
- This makes a lot of difference. From this fact arises the need for TCP and UDP to perform complete checksum to verify the data integrity.
- IP does not always run over ethernet
- IP does not checksum the data
- TCP packets can be reassembled incorrectly from IP packets and fragments that each have perfect checksums
- Even if reassembled correctly, software or other errors could be introduced in the layers between IP and TCP
- Every time a new header is introduced there is more to checksum, and the new layer can’t see the header bits of the layer below.

Q4 b) Differentiate between TCP and UDP [4]
Ans:

<table>
<thead>
<tr>
<th></th>
<th>UDP</th>
<th>TCP</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Datagram Protocol</td>
<td>Transmission Control Protocol</td>
<td></td>
</tr>
<tr>
<td>Message oriented protocol</td>
<td>Byte oriented protocol</td>
<td></td>
</tr>
<tr>
<td>Preserve message boundaries</td>
<td>Does not Preserve message boundaries</td>
<td></td>
</tr>
<tr>
<td>Unreliable</td>
<td>Reliable</td>
<td></td>
</tr>
<tr>
<td>No congestion and flow control</td>
<td>Have congestion and flow control</td>
<td></td>
</tr>
<tr>
<td>Each message follows different route so no sequencing</td>
<td>Each message follows same route so have in sequence data delivery</td>
<td></td>
</tr>
<tr>
<td>Port no 17</td>
<td>Port no 6</td>
<td></td>
</tr>
</tbody>
</table>
Q5 a) I have a subnet mask 255.255.255.248 set up in my machine with IP 10.5.5.20? What IP address should I ping to, so that I get response from all machines on my LAN subnet. [5]

Ans:
Step 1: convert IP 10.5.5.20 into Binary Value
00001010000001010000010100010100

Step 2: subnet mask 255.255.255.248 into Binary value
11111111111111111111111111111000

1 Indicate Network bit and 0 indicate Client bit (29 bits are at network side and 3 bits are at client side)

Step 3: Anding IP and Subnet mark

\[
\begin{array}{c}
00001010000001010000010100010100 \\
11111111111111111111111111111000 \\
\hline \\
00001010000001010000010100010000
\end{array}
\]

Decimal Value after Anding

10.5.5.16 (Network Address)

3 bits are at client side means \(2^3 = 8\) (i.e. 0 to 7). By adding this value in Network address we get 10.5.5.23 Broadcast Address.

10.5.5.23 IP address should ping to, so that I get response from all machines on my LAN subnet.

Q5b) Explain distance vector routing algorithm. What are advantages and disadvantages of DVR?

Ans:
Distance Vector Routing Algorithm: A distance-vector routing protocol requires that a router inform its neighbors of topology changes periodically. The term distance vector refers to the fact that the protocol manipulates vectors (arrays) of distances to other nodes in the network.

Distance Vector Routing Algorithm Working
Step 1) Initialization

- At the beginning, each node know the cost of itself and its immediate neighbor.[those node directly connected to it.]
- The distance of any entry that is not a neighbor is marked as infinite(unreachable).
Step 2) Sharing
- Idea is to share the information between neighbors.
- The node C does not know the distance about D, but C’s neighbor node A and node B does.
- If node A and B share it routing table with C, node C can also know how to reach node D.

Step 3) Updating in distance vector routing
- When node C get routing tables from A and B, it will compare the distance of both nodes to go to D, Whose ever distance is less, node C will update the entry in routing table according to it.

Advantages of DVR:
- Simple approach and easy to use, implement and maintain and does not require High-level knowledge to deploy.
- Moreover, it does not demand high bandwidth level to send their periodic updates as the size of the packets are relatively small.
- Furthermore, distance vector protocols do not require a large amount of CPU resources or memory to store the routing data.

Disadvantages of DVR:
- It is slower to converge than Link State.
- It is at risk from the count-to-infinity problem.
- It creates more traffic than Link State since a hop count change must be propagated to all routers and processed on each router.

Q 6a) Describe importance and working of ARP protocol? What is ARP Cache? [5]

Ans: **Importance of ARP:** ARP is used for mapping a network address (e.g. an IPv4 address) to a physical address like an Ethernet address (also named a MAC address). The purpose of Address Resolution Protocol (ARP) is to find out the MAC address of a device in your Local Area Network (LAN), for the corresponding IPv4 address, which network application is trying to communicate.
Working of Address Resolution Protocol (ARP)

Step 1: When a source device wants to communicate with another device, the source device checks its Address Resolution Protocol (ARP) cache to find if it already has a resolved MAC Address of the destination device. If it is there, it will use that MAC Address for communication. To view your Local Address Resolution Protocol (ARP) cache, open Command Prompt and type command "arp -a" (without double quotes using Windows Operating Systems).

Step 2: If ARP resolution is not there in local cache, the source machine will generate an Address Resolution Protocol (ARP) request message, it puts its own data link layer address as the Sender Hardware Address and its own IPv4 Address as the Sender Protocol Address. It fills the destination IPv4 Address as the Target Protocol Address. The Target Hardware Address will be left blank, since the machine is trying to find that.

Step 3: The source broadcast the Address Resolution Protocol (ARP) request message to the local network.

Step 4: The message is received by each device on the LAN since it is a broadcast. Each device compares the Target Protocol Address with its own Protocol Address (IPv4 Address). Those who do not match will drop the packet without any action.

Step 5: When it will find a match and will generate an Address Resolution Protocol (ARP) reply message.

ARP Cache: The ARP Cache is a collection of ARP entries (mostly dynamic) that are created when a hostname is resolved to an IP address and then an IP address is resolved to a MAC address (so the computer can effectively communicate with the IP address).

Q6b) An organization is granted the block 202.16.170.0/24. The administrator wants to create 24 subnets. [5]

i. Find the subnet mask.
ii. Find the number of addresses in each subnet.
iii. Find the first and last address in each subnet 1
iv. Find the first and last address in each subnet 24.

Ans:

- No of addresses for whole network are \( N = 2^{32-24} = 2^8 = 256 \)
  First Address in network is- 202.16.170.0/24
  Last address in network is- 202.16.170.255/24
- As 24 subnet has to design so we divide no of address with no of sub networks i.e. 256/24= 10. That means each network has 10 addresses.
- New subnet mask will be:
  For 24 network 5 bits are required (\( \log_2 24 = 5 \)) to add to /24 masks, so new mask will be 24+5 = /29.

<table>
<thead>
<tr>
<th>Sub network</th>
<th>First Address</th>
<th>Sub network</th>
<th>Last Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>202.16.170.0/29</td>
<td>24</td>
<td>202.16.170.255/29</td>
</tr>
</tbody>
</table>